**Azure Dynasty:**

**A PyGame RPG**

***Design***



Dominic Celiano

Dr. Bower, T1A

CS 210

11/19/14

**Program Summary**

This python program will be a two-dimensional role-playing game (RPG) in which the user will create a character and control it, moving through a two-dimensional world. The game view will be set as a birds-eye view, looking down onto the world and the character. The PyGame module will be used to create the game window and draw/animate all of the necessary items on the game screen, as well as get user input.

Azure Dynasty will be set in medieval times, and the main character will play the role of a duke/knight. This duke will go around killing monsters, leveling up and gaining better skills and attributes as it does so. It will be possible for the user to save progress and return to play the game at a later time.

**Requirements**

-Will display a splash screen when the game file is first started up.

-Will incorporate a main character to be controlled using the keyboard and/or mouse buttons.

-Will allow the user to save progress in their game to return to it at a later time.

-Will have monsters the main character can kill to level up and increase its skills.

-May also incorporate shops to purchase new equipment for the main character.

-May have bosses for the main character to face and attempt to defeat.

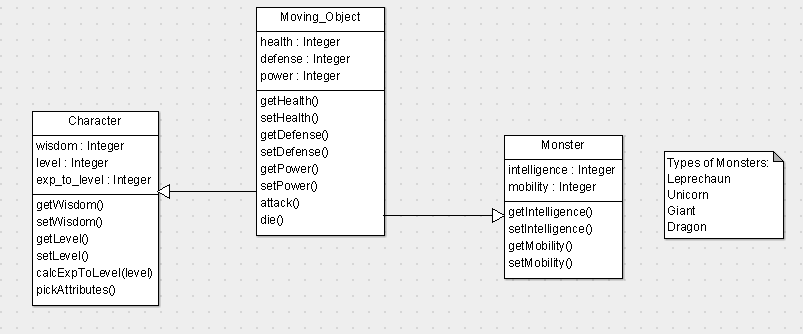
-May have different levels for the character to advance to.

**General Program Layout**

As with most games, this game will have a game loop (i.e. While True), where the drawing and clearing of the surface will actually be done. There will be functions and class methods called inside this game loop, which will be defined elsewhere in the program. The various parts of the screen and any other constant variables will also be initiated before the game loop. In order to separate the functionality of the character class/methods and the actual rendering of the game, the main character class may be put in a separate .py file. In order to save and load games, data structures will be serialized into a pickle file or stored in a .txt file.

**XML Class Diagram**

Below is the class setup that will be used for the main character and the monsters that the main character will attempt to kill. The attributes for each are listed, and can be changed as necessary.



**Function Descriptions**

**main():** The function where the screen will be initialized, all the variables set up, and the game loop started.

**runGame():** This function will contain the actual contents of the game loop; the main function will call this function when it calls the game loop.

**drawSurface():** This function will take care of drawing the surface and its mapped background, as well as anything else that needs to be drawn on the surface.

**gainLevel():** The character gains a level and consequently increases its attributes

**Attack():** The main character or monster subjects its enemy to random damage.

**calcExpToLevel(level):** Given what level the character currently is, this function calculates how much more experience will be needed to reach the next level

**pickAttributes():** This function allows the user to pick the attributes they want to advance on the character once it levels up.

**moveUp/Down/Left/Right():** Will be used to move the main character around depending on keys pressed, or randomly move around the monsters.

**drawMonster():** Draws the monster at a different spot on the screen as it moves around.

**die():** Once the main character or a monster is killed, it is cleared from the surface.